





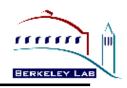
History Objects

ATLAS Software Workshop Architecture Session

Dec 1 2003



Purpose



- Provide provenance of data objects to be able to fully recreate conditions under which they were produced.
- This entails recording all aspects of the job, such as initial run time configurations, default initializations, software release information, random number seeds, hardware specific options, OS, etc.
- Since there will be a lot of History Objects, want to minimize redundant and derivable data as much as possible.





History Objects



> Job History

- date/time
- os, cpu, hardware environment
- ATLAS release
- Algorithms created/used
- Services created/used
- jobOptions
- created by the HistorySvc during initialization

> Service History

- properties
- identifier / version
- pointer to Job History
- created by ServiceManager when a new service is created, registered with HistorySvc via abstract interface



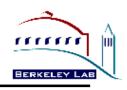
History Objects



- > (sub)Algorithm History
 - properties
 - identifier / version
 - list of subAlgorithms it creates
 - pointer to Job History
 - created by Algorithm base class via abstract interface to HistorySvc
- > AlgTools History
 - similar to Algorithms
- > Data History
 - pointer to parent Algorithm History object
 - ClassID/key
 - store identifier
 - created by StoreGate during a record via abstract interface to HistorySvc



History Service

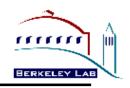


- Management of History Objects is performed by a History Service, accessed via the IHistorySvc abstract interface, located in GaudiKernel.
- Permits decoupling of HistorySvc from Gaudi and Athena.
- Provides access to History Objects, and associated properties, allowing users to trace provenance of an object to its parent Algorithm, Job, etc.





Implementation



- Implementation is transparent to users. Base classes of Algorithm, Service Locater, StoreGate, etc, are instrumented to create and update History Objects when appropriate.
- Will require a new StoreGate method: update(...) to replace record(...) when updating objects.
- > Job/Service/Algorithm history objects are kept in their own store.
- Data History objects are kept in the main store.
- > This makes persistification easier, as converters are triggered at the appropriate instant.